NPS55Hh71121A

United States
Naval Postgraduate School

ITERATIVE DETERMINATION OF PARAMETERS
FOR AN EXACT PENALTY FUNCTION

by

James K. Hartman

December 1971

Approved for public release; distribution unlimited.

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral A. S. Goodfellow, USN
Superintendent

M. U. Clauser
Provost

ABSTRACT

As an approach to solving nonlinear programs, we study a class of functions known to be exact penalty functions for a proper choice of the parameters. The goal is to iteratively determine the correct parameter values. A basic algorithm has been developed. We have proved that this algorithm converges for concave programs, and in the limited computational tests performed to date it has always converged for nonconcave programs also. Suggestions for continuing the work are given.

Prepared by:

TABLE OF CONTENTS

I. Introduction.

In recent years the technique of solving nonlinear programming problems using penalty functions has been extensively developed. The usual strategy is to convert a constrained nonlinear program to a sequence of unconstrained optimizations of "penalty functions." These penalty functions are constructed so that the sequence of unconstrained optimal solutions approaches the constrained optimal solution for the original nonlinear program. (See, for example, Fiacco and McCormick's book [3].)

This strategy may involve severe computational difficulties. Murray [6] has shown for several penalty function methods that the successive unconstrained problems become increasingly ill conditioned (and hence difficult to optimize) since a parameter must approach infinity to force the unconstrained optima to approach the solution of the nonlinear program.

An alternate strategy is to try to find a single function, called an "exact penalty function" such that the unconstrained optimal solution to the exact penalty function is exactly the optimal solution to the nonlinear program. Then a single unconstrained optimization of the exact penalty function will solve the nonlinear program. For the equality constrained problem, Fletcher [4] has found a continuously differentiable exact penalty function with a single parameter for which a correct parameter choice is not difficult. Similar results have been developed for the inequality constrained nonlinear program

by Zangwill [9], Pietrzykowski [8], and Evans et al [2]. In each of these cases, however, the exact penalty functions studied are not continuously differentiable. In fact, derivatives fail to exist along all constraint boundaries, and this is critical since in most nonlinear programs the optimal solution will occur at the boundary of the feasible region. The problem of finding a continuously differentiable exact penalty function for inequality constrained nonlinear programs is still not completely solved.

In this paper we examine a class of functions which are known to be continuously differentiable exact penalty functions *if* the parameters are chosen correctly. We then consider iterative procedures for selecting the proper parameter values. The result is a solution strategy which involves a sequence of unconstrained optimizations whose solutions converge to the optimal solution of the original nonlinear program. Thus the goal of only having to perform a single unconstrained optimization is not achieved. However, the successive unconstrained optimizations do not become increasingly ill-conditioned since moderate values of all parameters involved suffice to give a solution to the original nonlinear program. Hence, this approach may offer an improvement over conventional penalty function schemes.

II. The Method.

Throughout this discussion we will consider the following inequality constrained nonlinear program:

$$\text{(NLP)} \quad \text{maximize} \quad f(x)$$
$$\text{subject to} \quad g_i(x) \le 0 \qquad i = 1,\dots,m \tag{1}$$
$$x \in E^n$$

with Lagrangian function

$$L(x,\lambda) = f(x) - \sum_{i=1}^{m} \lambda_i g_i(x).$$

Suppose f and g_i $(i = 1,\dots,m)$ are twice continuously differentiable. For simplicity it will be assumed that NLP has at least a local solution x^*, and that a constraint qualification applies so that the Kuhn-Tucker necessary conditions hold at x^* with optimal multiplier vector $\lambda^* \in E^m$. Let S denote the feasible region for NLP(1): $S = \{x \in E^n \mid g_i(x) \le 0,\ i = 1,\dots,m\}$. Let $E_+^m = \{\lambda \in E^m \mid \lambda_i \ge 0,\ i = 1,\dots,m\}$.

a) The exact penalty function.

Consider the function of $x \in E^n$, $k \in E_+^1$, $\lambda \in E_+^m$

$$P(x,k,\lambda) = f(x) - (1/k) \sum_{i=1}^{m} \lambda_i [\exp(k\, g_i(x)) - 1] \tag{2}$$

Note that for $\lambda_i > 0$, $k > 0$ the exponential term imposes a penalty for values of x which violate the constraint $g_i(x) \leq 0$.

The following theorem has been proved by Gould and Howe [5]:

<u>Theorem 1</u>. Suppose $x^* \varepsilon E^n$ is a local solution for the nonlinear program (1) and $\lambda^* \varepsilon E_+^m$, are multipliers such that

A1. $\nabla_x L(x^*,\lambda^*) = 0$ (∇_x denotes the gradient vector with respect to x)

A2. $g_i(x^*) \leq 0$ $i = 1,\ldots,m$

A3. $\sum_{i=1}^{m} \lambda_i^* g_i(x^*) = 0$

A4. $z^T \nabla_x^2 L(x^*,\lambda^*) z < 0$ for each nonzero $z \varepsilon E^n$ satisfying $z^T \nabla_x g_i(x^*) = 0$ for all i such that $\lambda_i^* > 0$

A5. $\lambda_i^* > 0$ for every i such that $g_i(x^*) = 0$.

Then there exists a $K > 0$ such that for every $k^* > K$, x^* is an unconstrained isolated local maximizer of $P(x,k^*,\lambda^*)$.

Note that conditions A1, A2, A3 of the theorem are just the Kuhn Tucker necessary conditions, while, in the presence of the nondegeneracy condition A5, the conditions A1, A2, A3, A4 are equivalent to the second order sufficiency conditions given by Fiacco and McCormick [3].

Theorem 1 implies that if the optimal Lagrange multipliers λ^* are known, and if k^* is chosen large enough, then the function $P(x,k^*,\lambda^*)$ can be considered to be an exact penalty function. It

is easy to see that P has as many orders of continuous derivatives as the functions f and g_i have. Of course, in solving a nonlinear program, we do not know the optimal multipliers in advance.

In this paper we will address the question of how to find the optimal multipliers λ^* using an iterative scheme. Throughout we will assume that k^* is fixed sufficiently large (note that in the computations reported in section V, moderate values of k^* were always sufficient).

b. Recognizing an optimal solution.

Suppose that some value of λ has been chosen. If this λ is the optimal Lagrange multiplier vector λ^*, then given the conditions of theorem 1, there exists a local maximum x^* of $P(x,k,\lambda)$ which solves NLP (1). The following lemmas will enable us to recognize when this has occurred.

Lemma 2. Suppose x^* is a local max for $P(x,k,\lambda)$ (for some fixed values of $\lambda \geq 0$, $k > 0$) then the Kuhn Tucker conditions are satisfied at (x^*,λ) if and only if x^* is a local optimal solution to NLP.

Proof. Necessity of the Kuhn Tucker conditions for a solution to NLP has been assumed from the beginning. To prove that the conditions are sufficient we argue as follows. Since x^* is a local maximum for $P(x,k,\lambda)$, $\exists\ \delta > 0$ such that $\forall\ x \in N_\delta(x^*)$ we know

$$P(x^*,k,\lambda) \geq P(x,k,\lambda) \tag{3}$$

Let $\hat{x}$ be any point in $N_\delta(x^*) \cap S$. We want to show $f(x^*) \geq f(\hat{x})$.

From (3)

$$P(x^*,k,\beta) \geq P(\hat{x},k,\beta)$$

Then using the definition of P in (2),

$$f(x^*) - (1/k) \sum_{i=1}^{m} \lambda_i[\exp(k\, g_i(x^*)) - 1] \geq$$

$$f(\hat{x}) - (1/k) \sum_{i=1}^{m} \lambda_i[\exp(k\, g_i(\hat{x})) - 1] \tag{4}$$

Now

$$\sum_{i=1}^{m} \lambda_i[\exp k\, g_i(x^*) - 1] = 0 \tag{5}$$

since by complementary slackness if $\lambda_i > 0$ then $g_i(x^*) = 0$.

Also

$$\sum_{i=1}^{m} \lambda_i[\exp k\, g_i(\hat{x}) - 1] \leq 0 \tag{6}$$

since $\lambda_i \geq 0$ and feasibility of $\hat{x}$ implies $\exp(k\, g_i(\hat{x})) - 1 \leq 0$ $\forall i$.

Combining (4), (5), and (6) we have

$$f(x^*) \geq f(\hat{x}).$$

Thus since x^* is feasible, x^* is a local optimal solution for NLP (1).

$\square$

In the above sufficiency proof, only part of the Kuhn Tucker conditions were required. In particular we did not need to use the fact that $\nabla_x L(x^*,\lambda) = 0$. Thus a more economical way of writing Lemma 2 is as follows:

Lemma 3. For fixed $\lambda \in E^m_+$, $k > 0$, if

i) x^* is a local unconstrained max for $P(x,k,\lambda)$

ii) $\lambda_i g_i(x^*) = 0 \qquad i = 1,\ldots,m$

iii) $g_i(x^*) \leq 0 \qquad i = 1,\ldots,m$

Then x^* is a local maximum for the nonlinear program NLP (1).

Proof. Same as for Lemma 2.

This result is interesting for two reasons. First, it provides a condition (x^* maximizes $P(x,k,\lambda)$) under which the Kuhn Tucker Conditions are necessary and sufficient for x^* to be a local max of NLP (1). This is true for any nonlinear program satisfying a constraint qualification--no convexity properties are required.

Second, Lemma 3 provides a termination condition for an iterative scheme of determining the optimal multipliers λ^*.

c) Iterative determination of optimal multipliers.

Suppose λ is not an optimal multiplier vector. Then we face the problem of choosing better values for the multipliers. In this paper we will consider a single iterative scheme for improving the values of λ. The procedure is as follows:

1. Choose $\lambda^1 \in E^m_+$. Set $s = 1$ as an iteration counter. Fix $k > 0$.

2. Let $x^s \in E^n$ be a local maximum for $P(x,k,\lambda^s)$ obtained by starting from x^{s-1} and using an unconstrained optimizer. If the conditions ii, iii of Lemma 3 are satisfied, STOP: x^s is optimal.

3. Let $\lambda^{s+1} \in E^m_+$ have i^{th} component

$$\lambda_i^{s+1} = \lambda_i^s \exp(k\, g_i(x^s)) \tag{7}$$

Replace s by $s + 1$ and go to step 2.

An intuitive justification for the correction in (7) can be given for an NLP with one constraint. In this case, if λ^s is larger than the optimal λ^*, then we expect the maximizing x to strictly satisfy the constraint, $g(x) < 0$. Hence

$$\lambda^{s+1} = \lambda^s \exp k\, g(x) < \lambda^s$$

so λ is made smaller for the next iteration. Similarly if λ^s is smaller than the optimal λ^* then we expect the maximizing x to violate the constraint. Thus in this case

$$\lambda^{s+1} = \lambda^s \exp k\, g(x) > \lambda^s$$

and again λ is corrected in the appropriate direction. Of course, when there are several constraints, the interaction between them makes the above intuitive reasoning less clear. Mathematical justification for the correction (7) will be given later.

Since λ at each iteration is computed from (7) and since $\exp k\, g_i(x) > 0$ for any x, the resulting λ_i^{s+1} will never equal zero unless $\lambda_i^1 = 0$.

Practically, however, if $g_i(x) < 0$, $\exp k\, g_i(x)$ may be very small, so that after several cycles λ_i^{s+1} will be indistinguishable from zero on a computer with finite word length. Note also that $\lambda_i \geq 0$ is guaranteed automatically by the formula (7) at all iterations.

Suppose λ^s has been given, and that x^s maximizes $P(x,k,\lambda^s)$. If x^s solves NLP (1) then the Kuhn Tucker conditions will be satisfied and the algorithm will stop. If, however, x^s is not optimal, then by Lemma 2 the Kuhn Tucker conditions cannot be satisfied. In this case the method computes new multipliers using (7)

$$\lambda_i^{s+1} = \lambda_i^s \exp(k\, g_i(x^s)) \tag{8}$$

Note that if $\lambda_i^s \neq 0$ and if $g_i(x^s) \neq 0$ then the new multiplier λ_i^{s+1} will be different from λ_i^s. The only conditions in which $\lambda_i^{s+1} = \lambda_i^s$, $i = 1,\ldots,m$ can occur are

1. x^s is feasible and $\lambda_i^s g_i(x^s) = 0\ \forall i$ (but then by Lemma 3 x^s is optimal)

or 2. for some i $g_i(x^s) > 0$ while $\lambda_i^s = 0$. In this case, when $\lambda_i^s = 0$ the penalty term for the i^{th} constraint drops out of the P function. We can avoid this by resetting λ_i^{s+1} to some positive quantity if $g_i(x^s) > 0$ and $\lambda_i^s = 0$.

(In our computations to date it has never been necessary to do this, but for finite word length computations it is a possibility.)

If the above precaution is taken, then the algorithm will never fail to generate a different λ vector at each iteration unless an optimal solution has been reached.

III. Dual Feasibility.

The formula (7) chosen to correct the Lagrange multipliers is motivated by the following duality considerations. Note that

$$\nabla_x L(x^s,\lambda^{s+1}) = \nabla_x f(x^s) - \sum \lambda_i^{s+1} \nabla_x g_i(x^s)$$

$$= \nabla_x f(x^s) - \sum \lambda_i^s \exp(k\ g_i(x^s))\nabla_x g_i(x^s)$$

$$= \nabla_x\ P(x^s,k,\lambda^s) = 0 \tag{9}$$

since $P(x,k,\lambda^s)$ has an unconstrained max at x^s. Hence at each stage of the algorithm, the point (x^s,λ^{s+1}) gives a stationary point for the Lagrangian function. The method can thus be viewed as one which chooses λ^{s+1} to minimize the error in the Kuhn Tucker condition $\nabla_x L(x,\lambda) = 0$.

If NLP (1) is a concave program, that is if f is concave and each g_i is convex, then the Lagrangian function is concave in x for fixed λ. In this case (9) above and $\lambda^{s+1} \geq 0$ show that the point (x^s,λ^{s+1}) is feasible for the Wolfe dual problem to NLP

$$\text{(Dual)} \qquad \begin{array}{ll} \text{maximize} & L(x,\lambda) \\ \text{subject to} & \nabla_x L(x,\lambda) = 0 \\ & \lambda \geq 0 \end{array} \tag{10}$$

Since the Lagrangian is concave in x, the stationary point at (x^s,λ^{s+1}) must be a maximum of $L(x,\lambda^{s+1})$ with respect to x. We will use this fact in proving convergence of the method for concave programs.

IV. Convergence.

In section III it was shown that the formula (7) will always generate a new vector of multipliers λ unless the optimum solution has been reached. In this section we show that this sequence of vectors (and hence the sequence of x vectors generated by them) is an improving sequence, so that the method will converge. We have been able to demonstrate convergence only for concave programs, but in the limited computational tests performed to date, the method has always converged for non-concave programs also.

It should be noted that for concave programs, the ordinary Lagrangian function is an exact penalty function if the optimal multipliers are known, and iterative methods for obtaining the multipliers are also available. For non-concave programs, however, methods using the Lagrangian as a penalty function are known not to converge. (In particular they will fail on some of the simple examples presented in section V.)

The advantage of the P function for nonconcave programs may be explained by its saddlepoint properties. A function $F(x,\lambda)$ is said to have a saddlepoint at (x^*,λ^*) if

$$F(x,\lambda^*) \leq F(x^*,\lambda^*) \leq F(x^*,\lambda) \tag{11}$$

for all $x \in E^n$, $\lambda \in E^m_+$. For concave programs both the Lagrangian $L(x,\lambda)$ and the exact penalty function $P(x,k,\lambda)$ (with k held fixed) are concave in x for fixed $\lambda \geq 0$ and linear in λ for fixed x. Hence, a saddlepoint exists at the optimal solution to the NLP (1).

If the NLP (1) is not a concave program, then the Lagrangian function will generally not have a saddlepoint at the solution (x^*,λ^*) to the NLP. However, as shown by Gould and Howe [5], the P function will always have a local saddlepoint at (x^*,λ^*) (for k chosen sufficiently large and under the conditions of Theorem 1).

To establish convergence, we first prove the following:

<u>Lemma 4</u>. Let x^s, λ^s be generated by the method of section IIc. for $s = 1,2,\ldots,$ and suppose the complementary slackness condition $\lambda_i^s g_i(x)^s = 0$ $(i = 1,\ldots,m)$ is not satisfied. Then

$$L(x^s,\lambda^{s+1}) < L(x^s,\lambda^s) \qquad s = 1,2,\ldots .$$

<u>Proof</u>. $L(x^s,\lambda^{s+1}) = f(x^s) - \sum_{i=1}^{m} \lambda_i^{s+1} g_i(x^s)$

$$= f(x^s) - \sum_{i=1}^{m} \lambda_i^s \exp(k\, g_i(x^s)) g_i(x^s) \tag{12}$$

from the definition of the Lagrangian and from (7). Now $\lambda_i^s \geq 0$ for all $i = 1,\ldots,m$. If $g_i(x^s) > 0$ then

$$\exp(k\, g_i(x^s)) g_i(x^s) > g_i(x^s) > 0 \tag{13}$$

while if $g_i(x^s) < 0$, then

$$g_i(x^s) < \exp(k\, g_i(x^s)) g_i(x^s) < 0. \tag{14}$$

Finally, if $g_i(x^s) = 0$ then

$$g_i(x^s) = \exp(k\ g_i(x^s))g_i(x^s) = 0 \tag{15}$$

In each case (13), (14) and (15),

$$\lambda_i^s g_i(x^s) \leq \lambda_i^s \exp(k\ g_i(x^s))g_i(x^s) \tag{16}$$

and since complementary slackness does not hold, the inequality in (16) is strict for at least one i.

Hence,

$$f(x^s) - \sum_{i=1}^{m} \lambda_i^s \exp(k\ g_i(x^s))g_i(x^s) < f(x^s) - \sum_{i=1}^{m} \lambda_i^s g_i(x^s)$$

that is,
$$L(x^s,\lambda^{s+1}) < L(x^s,\lambda^s) \tag{17}$$

which completes the proof.

□

Lemma 5. If NLP (1) is a concave program, then for all s,

$$L(x^{s+1},\lambda^{s+1}) \leq L(x^s,\lambda^{s+1}) \tag{18}$$

Proof. $\nabla_x L(x^s,\lambda^{s+1}) = 0$ by definition of λ^{s+1} as shown in section III. If NLP (1) is a concave program, then $L(x,\lambda)$ is a concave function of x for fixed $\lambda \geq 0$. Thus the stationary point (x^s,λ^{s+1}) must be a maximum of L with respect to x. Hence

$$L(x,\lambda^{s+1}) \leq L(x^s,\lambda^{s+1})$$

for all $x \in E^n$, and letting $x = x^{s+1}$ completes the proof.

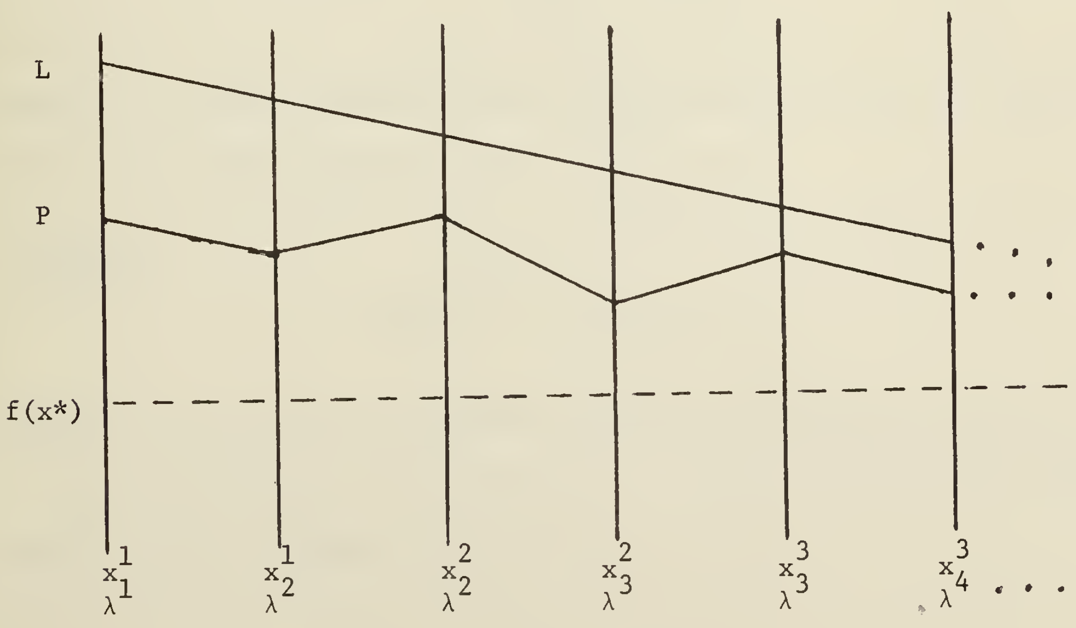
□

As a consequence of lemmas 4 and 5 we know that for a concave program

$$L(x^1,\lambda^1) > L(x^1,\lambda^2) \geq L(x^2,\lambda^2) \quad \ldots \geq L(x^s,\lambda^s) > L(x^s,\lambda^{s+1}) \geq \ldots \qquad (19)$$

as long as a solution has not been reached. Since the points (x^s,λ^{s+1}) are feasible for the dual (10) and since the dual objective is always greater than the primal objective, we know that the entire decreasing sequence (19) is bounded below by the optimal value $f(x^*)$ for NLP (1). For a concave problem the method can thus be seen to be a descent method applied to the dual problem.

It is interesting to observe that the penalty function P which we are actually optimizing in the method increases when x values are changed. It is also true that $\forall\, x \in E^n$, $\forall\, \lambda \in E^m_+$, $L(x,\lambda) \geq P(x,k,\lambda)$ so that the following diagram can be drawn to describe the progress of the method for concave programs.



Convergence of the method can be proved for concave programs using a standard convergence theorem from Zangwill [10]. The algorithmic map is given by

$$A(x^s,\lambda^s) = (x^s,\lambda^{s+1}) \tag{20}$$

$$A(x^s,\lambda^{s+1}) = (x^{s+1},\lambda^{s+1}) \tag{21}$$

and the Lagrangian function $L(x,\lambda)$ serves as the adaption function. For $\lambda^{s+1} > 0$, the map A is continuous and hence a closed map. Then (19) shows that the adaption function improves at every iteration (with strict improvement every second step) so that convergence follows from Zangwill's results. This can be summarized by

Theorem 6. If NLP (1) is a concave program, then the method of section IIc. will converge to the optimal solution (x^*,λ^*).

Proof. (as above).

V. Computational Experience.

An experimental computer program has been written to gain experience with the basic method of section II. The program is written in FORTRAN IV and the experiemental runs were done on the IBM 360 at the Naval Postgraduate School. In its current state the program is intended to test the efficacy of our strategy for iteratively determining the optimal multiplier vector. No attempt has been made to optimize the code either in terms of running time or function evaluations although there are many opportunities for doing so at the expense of more complicated coding.

In the process of debugging the program several small problems were solved. Efforts are now continuing with larger and more complex problems.

Problem 1.

$$\begin{aligned} \max \quad & -x_1 - x_2 \\ \text{subject to} \quad & x_1^2 - x_2 \le 0 \\ & -x_1 \le 0 \end{aligned}$$

This is a concave program given as an example by Fiacco and McCormick [3]. The optimal solution is at $x_1 = x_2 = 0$.

Problem 2.

$$\begin{aligned} \max \quad & x^3 \\ \text{subject to} \quad & x - 2 \le 0 \end{aligned}$$

Problem 2 is not a concave program, but it has a single local max at $x = 2$. The KTC are also satisfied at $x = 0$. The Lagrangian function does not have a saddlepoint at $x = 2$.

<u>Problem 3</u>.

$$\begin{aligned} \max \quad & x_1 x_2 \\ \text{subject to} \quad & x_1 + x_2^2 - 1 \le 0 \\ & -x_1 - x_2 \le 0 \end{aligned}$$

Problem 3 is not concave, but it has a single local max at $x_1 = 2/3$, $x_2 = \sqrt{3}/3$.

<u>Problem 4</u>.

$$\begin{aligned} \max \quad & 2\,x_1 + x_2^2 \\ \text{subject to} \quad & 2\,x_1 + x_2 \le 2 \\ & x_1 \ge 0, \quad x_2 \ge 0 \end{aligned}$$

Problem 4 has a local max at $x_1 = 1$, $x_2 = 0$ and the global max at $x_1 = 0$, $x_2 = 2$. As will be seen, the method may converge to either of these depending on the starting point.

<u>Problem 5</u>.

$$\begin{aligned} \max \quad & x_1 x_2 x_3 \\ \text{subject to} \quad & x_1 + 2\,x_2 + 2\,x_3 \le 72 \\ & 0 \le x_1 \le 42 \\ & 0 \le x_2 \le 42 \\ & 0 \le x_3 \le 42 \end{aligned}$$

Rosenbrock's post office parcel problem has optimal solution $x_1 = 24$, $x_2 = 12$, $x_3 = 12$.

Problem 6.

$$\min \quad (x_1-1)(x_1-2)(x_1-3) + x_3$$

$$\text{subject to} \quad x_1^2 + x_2^2 - x_3^2 \le 0$$

$$4 - x_1^2 - x_2^2 - x_3^2 \le 0$$

$$x_3 - 5 \le 0$$

$$x_1 \ge 0, \quad x_2 \ge 0, \quad x_3 \ge 0$$

Problem 6, formulated at Research Analysis Corporation, has a nonconvex objective function and nonconvex feasible region. The optimal solution occurs at $x_1 = 0$, $x_2 = \sqrt{2}$, $x_3 = \sqrt{2}$.

Problem 7.

$$\min \quad \sum_{j=1}^{5} e_j x_j + \sum_{i=1}^{5} \sum_{j=1}^{5} x_i c_{ij} x_j + \sum_{j=1}^{5} d_j x_j^3$$

$$\text{subject to} \quad \sum_{j=1}^{5} a_{ij} x_j \ge b_i \qquad i = 1,\ldots,10$$

$$x_j \ge 0 \qquad j = 1,\ldots,5$$

Problem 7, with coefficients given in Mylander et al [7] is due to the Shell Development Company. It is a convex program. Problem 7 and its dual Problem 8 were used as test problems in Colville's comparison of NLP codes [1].

<u>Problem 8</u>.

$$\max \sum_{j=1}^{10} b_j y_j - \sum_{i=1}^{5} \sum_{j=1}^{5} x_i c_{ij} x_j - 2 \sum_{i=1}^{5} d_i x_i^3$$

$$\text{subject to} \quad \sum_{j=1}^{10} a_{ji} y_j \le e_i + 2 \sum_{j=1}^{5} c_{ji} x_j + 3\, d_i x_i^2 \qquad i = 1,\ldots,5$$

$$x_i \ge 0 \qquad i = 1,\ldots,5$$

$$y_j \ge 0 \qquad j = 1,\ldots,10$$

Problem 8 is the dual of problem 7. It is not a concave program, and the feasible region is not a convex set.

Results of the algorithm as applied to these test problems are given in table 1. As a termination condition we required the Kuhn Tucker Conditions to be satisfied to within 10^{-6}, that is,

$$\sum_{i=1}^{m} \max(g_i(x),0)^2 \le 10^{-6}$$

$$\sum_{i=1}^{m} (\lambda_i g_i(x))^2 \le 10^{-6}$$

$$\sum_{j=1}^{n} \left(\frac{\partial L(x,\lambda)}{\partial x_j}\right)^2 \le 10^{-6}.$$

In all cases this led to solution values accurate to within 10^{-4} of the theoretical optimum. For convenience all λ_i were initially set to 10.0 for all problems.

Note that all problems were solved successfully for moderate values of k (5 or 10). In general the larger values of k gave

TABLE 1

Results of Computational Tests

Problem	Vari-ables	Con-straints	Initial x	k	No. iterations	
1	2	2	(1,1)	5	7	
			(1,1)	10	6	
2	1	1	(0)	5	7	
			(0)	10	5	
3	2	2	(1,1)	1	Did not converge	
			(1,1)	5	5	
			(1,1)	10	4	
4	2	3	(0,0)	5	17	(global)
			(0,0)	10	7	(global)
			(0.5,0)	10	7	(local)
5	3	7	(10,10,10)	5	4	
			(10,10,10)	10	4	
6	3	6	(0,0,3)	5	7	
			(0,0,3)	10	5	
7	5	15	(.1,.1,.2,.5,.5)	5	13	
			(.1,.1,.2,.5,.5)	10	10	
8	15	20	$x_j = 10^{-4} (j \neq 7)$			
			$x_7 = 60.0$	5	33	

faster convergence in terms of the number of unconstrained optimizations required, but it should be noted that if k increases, then the function $P(x,k,\lambda)$ will be harder to optimize due to the presence of sharp corners in the $\exp(k\, g_i(x))$ terms. In problem 4 the method converged either to a local max or to the global max depending on the initial x values.

VI. Conclusions and Extensions.

The computational results reported in section V indicate that our method has some promise as a tool for solving nonlinear programs. There are still, however, many improvements which must be made before routine use is possible.

The first problem with the basic method is that for each iteration reported in section V a complete unconstrained optimization is required. It would be better if the method could be modified so that revised λ estimates are made more frequently. We are currently investigating this problem.

The question of convergence for non-concave problems is not yet resolved. Further computational tests and theoretical work are planned to help answer this question.

The major goal which has been achieved is the development of a sequential unconstrained penalty function technique in which the successive unconstrained problems do not become increasingly ill conditioned. Hopefully, this will tend to alleviate the numerical problems encountered in other penalty function methods.

REFERENCES

[1] Colville, A. R., "A Comparative Study on Nonlinear Programming Codes," IBM Technical Report - 320-2947, June 1968.

[2] Evans, J. P., F. J. Gould and J. W. Tolle, "Exact Penalty Functions and Duality in Nonlinear Programming,"

[3] Fiacco, A. V. and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, Inc., New York, 1968.

[4] Fletcher, R., "A Class of Methods for Nonlinear Programming with Termination and Convergence Properties," in *Integer and Nonlinear Programming*, J. Abadie, ed., North Holland Publishing Company, Amsterdam, 1970.

[5] Gould, F. J. and S. Howe, "A New Result on Interpreting Lagrange Multipliers as Dual Variables," University of North Carolina Institute of Statistics, Mimeo series No. 738, January 1971.

[6] Murray, W., "Ill Conditioning in Barrier and Penalty Functions Arising in Constrained Nonlinear Programming," Princeton Symposium on Math Programming, 1967.

[7] Mylander, W. C., R. L. Holmes and G. P. McCormick, "A Guide to SUMT, Version 4," RAC-P-63, October 1971.

[8] Pietrzykowski, T., "An Exact Potential Method for Constrained Maxima," SIAM J. Numer. Analysis, Vol. 6, No. 2, June 1969.

[9] Zangwill, W. I., "Nonlinear Programming via Penalty Functions," Management Science, Vol. 13, No. 5, January 1967.

[10] Zangwill, W. I., *Nonlinear Programming: A Unified Approach*, Prentice Hall, Englewood Cliffs, New Jersey, 1969.

INITIAL DISTRIBUTION LIST

Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
Dean of Research Administration Code 023 Naval Postgraduate School Monterey, California 93940	1
Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
Dr. Thomas Varley Office of Naval Research Arlington, Virginia 22217	1
Library, Code 55 Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	5
Professor Donald P. Gaver Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
Professor G. T. Howard Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
Professor James K. Hartman Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	10

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Iterative Determination of Parameters for an Exact Penalty Function

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report

5. AUTHOR(S) (First name, middle initial, last name)

James K. Hartman

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1971	30	8

8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

As an approach to solving nonlinear programs, we study a class of functions known to be exact penalty functions for a proper choice of the parameters. The goal is to iteratively determine the correct parameter values. A basic algorithm has been developed. We have proved that this algorithm converges for concave programs, and in the limited computational tests performed to date it has always converged for nonconcave programs also. Suggestions for continuing the work are given.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Nonlinear Programming						
Penalty Functions						
Mathematical Programming						
Optimization						
Lagrange Multipliers						